



# DESARROLLO WEB

## BUENAS PRÁCTICAS EN EL DESARROLLO FRONTEND

# BUENAS PRÁCTICAS EN EL DESARROLLO FRONTEND: ACCESIBILIDAD Y OPTIMIZACIÓN DE RENDIMIENTO



Las buenas prácticas en desarrollo frontend abarcan un conjunto integral de metodologías, técnicas y principios que garantizan la creación de aplicaciones web robustas, accesibles, eficientes y mantenibles a largo plazo. La accesibilidad web no constituye una característica adicional sino un requisito fundamental que debe integrarse desde las etapas iniciales del desarrollo, siguiendo las pautas WCAG (Web Content Accessibility Guidelines) e implementando patrones de diseño inclusivo que benefician a todos los usuarios. Estas prácticas incluyen el uso correcto de marcado semántico, la implementación de navegación por teclado, el manejo apropiado del contraste de colores, y la provisión de alternativas textuales para contenido multimedia.

La optimización de rendimiento en aplicaciones frontend requiere una aproximación holística que considere desde la optimización de recursos hasta patrones de carga eficientes, impactando directamente en métricas críticas como Tiempo al Primer Byte, Primera Pintura con Contenido y Cambio de Diseño Acumulativo (Granados La Paz, 2023). Las técnicas modernas incluyen la implementación de carga diferida para imágenes y componentes, la utilización de división de código para reducir el tamaño inicial de paquetes JavaScript, y la optimización de la Ruta de Renderizado Crítica mediante la incrustación de CSS crítico y el diferimiento de recursos no esenciales. La adopción de formatos de imagen modernos como WebP y AVIF, junto con la implementación de imágenes adaptativas usando srcset y sizes, puede reducir significativamente los tiempos de carga mientras mantiene la calidad visual.

En aplicaciones empresariales de gran escala, como sistemas ERP o plataformas de gestión de recursos humanos, la implementación de buenas prácticas de accesibilidad y rendimiento resulta crítica para garantizar productividad e inclusión en el ambiente laboral. Por ejemplo, un sistema de gestión de inventario debe proporcionar navegación por teclado eficiente para usuarios que procesan grandes volúmenes de datos, implementar indicadores de progreso accesibles para operaciones de carga, y utilizar técnicas de virtualización para manejar listas con miles de elementos sin impactar el rendimiento. La correcta implementación de enlaces de salto, marcadores ARIA y descripciones textuales para gráficos de datos asegura que todos los empleados puedan utilizar efectivamente la herramienta.

Ejercicio práctico. Optimizar una página de listado de productos implementando técnicas avanzadas de rendimiento y accesibilidad.

- **Primero**, implementar carga diferida para imágenes utilizando Intersection Observer API: crear un observador que detecte cuando las imágenes entran en la ventana gráfica y reemplace el data-src por src, incluyendo un marcador de posición visual durante la carga.
- **Segundo**, optimizar la accesibilidad implementando navegación por teclado con tabindex apropiado, agregando aria-label descriptivos para botones de acción y utilizando role="button" para elementos no estándar que funcionan como botones.

- **Tercero**, implementar división de código cargando dinámicamente el JavaScript de funcionalidades avanzadas solo cuando se necesiten usando `import()` dinámico.
- **Cuarto**, medir el rendimiento usando Lighthouse y Web Vitals, identificando oportunidades de mejora en métricas como LCP y CLS.

El resultado es una página que carga rápidamente, proporciona excelente experiencia de usuario para personas con discapacidades, y mantiene rendimiento óptimo incluso con grandes cantidades de productos.